UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/980,761	04/15/2002	Michael R Krause	10003628-2	4254

22879 7590 11/29/2005

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
REILLY, SEAN M	
ART UNIT	**PAPER NUMBER**
2153	

DATE MAILED: 11/29/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 09/980,761	Applicant(s) KRAUSE ET AL.
	Examiner Sean Reilly	Art Unit 2153

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *28 September 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *2-47* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *2-47* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

This Office action is in response to Applicant's amendment and request for reconsideration filed on 9/28/2005. Claims 2-47 are presented for further examination. All independent claims have been amended.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

1. Claims 2-18, 22, 25-41, and 45 are rejected under 35 U.S.C. 103(a) as being unpatentable over Request for Comment 793 (Transmission Control Protocol, hereinafter RFC 793) and Mockapetris (Analysis of Reliable Multicast Algorithms for Local Networks, Paul Mockapetris) and Block et al. (U.S. Patent Number 6,192,417; hereinafter Block).

2. Regarding claim 2, RFC 793 discloses a distributed computer system comprising:

- a source endnode including:
    - a. a source process which produces message data (pg 7, last ¶ continued on pg 8 and pg 24, last ¶, first sentence);
    - b. a send work queue having work queue elements that describe the message data for sending (pg 24, last ¶ and pg 41, 3rd ¶, last sentence);
- destination endnode including:
    - a. a destination process (pg 7, last ¶ continued on pg 8);

b. a receive work queue having work queue elements that describe where to place incoming message data (pg 7, last ¶ continued on pg 8);

- communication fabric providing communication between the source endnode and the destination endnode (inherent; pg 7, last ¶ continued on pg 8); and
- an end-to-end context at the source endnode and the destination endnode storing state information to ensure the reception and sequencing of message data sent from the source endnode to the destination endnode thereby permitting reliable datagram service between the source endnode and the destination endnode (Section 2.6 beginning on pg 9).

However, RFC 793 fails to disclose 1) reliable *multicast* to a group of destination endnodes wherein the reliable multicast comprise a series of replicated unicasts to each endnode and 2) generating a completion event to the source process in response to an indication that a *predetermined percentage of destination endnodes in the multicast group have reliably received a selected amount of message data multicast* from the source endnode.

With regard to point 1, reliable multicasting to a group of recipients through a series of replicated packets was well known in the art at the time of invention, as evidenced by Mockapetris. In a related art Mockapetris teaches a reliable multicasting method where the sender sends a separate (replicated) message to each destination endnode and receives an acknowledgement of receipt from each endnode separately (pg 153 Simulation algorithms, 1st ¶). Mockapetris further discloses that the sender maintains the multicast group of destination endnodes as a list (pg 153 Simulation algorithms, 1st ¶). It would have been obvious to one of ordinary skill in the art at the time of invention to add the multicast functionality disclosed by

Mockapetris to the one-to-one transmission system disclosed by RFC 793 in order to provide a straightforward method for reliable one-to-many transmission (pg 153 Simulation algorithms, 1st ¶).

In the combined Mockapetris and RFC 793 system, all one-to-one transmission capabilities defined in the RFC 793 are expanded to a one-to-many (multicast) transmission capability since the one-to-many transmission is simply a series of replicated one-to-one transmissions to each multicast member endnode, as disclosed by Mockapetris (Cited above).

With regard to point 2, while RFC 793 discloses generating a completion event (TCP-to-user signals) for each endnode that acknowledges a received frame (pg 41, 5th ¶) neither RFC 793 nor Mockapetris specifically recited generating a completion event to the source process in response to an indication that a *predetermined percentage of destination endnodes in the multicast group have reliably received a selected amount of message data multicast* from the source endnode. Nonetheless it was widely known in the art at the time of the invention to notify application source processes that make multicast requests the status of such requests, as evidenced by Block. In a similar multicasting system, Block disclosed multicasting messages to a group of nodes (Col 9, lines 28-34). Like RFC 793 and Mockapetris, Block relies on acknowledgments from each node in the multicast group to determine which nodes have reliably received each multicast message (Col 15, lines 51-57 or Col 16, lines 8-11). In Block's system the source process that submits the multicast request can receive a completion event (notification) when a predetermined percentage of nodes (i.e. all nodes in the multicast group) have reliably received the multicast message (see inter alia Col 13, lines 40-43 and Col 16, lines 8-11, 18-19). This notification scheme allows the source process that submits a multicast request

to be notified of the multicast request status. Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to incorporate the completion event notification scheme, as disclosed by Block, within the combined RFC 793 and Mockapetris system, so that application processes which use the combined RFC 793 and Mockapetris multicast system are apprised of the status of their multicast requests.

3. Regarding claim 25, RFC 793 discloses a method of sending message data in a distributed computer system, the method comprising:

- producing message data with a source process at the source endnode (pg 7, last ¶ continued on pg 8 and pg 24, last ¶, first sentence);
- describing the message data for sending with work queue elements in a send work queue at the source endnode (pg 24, last ¶ and pg 41, 3rd ¶, last sentence);
- describing where to place incoming message data with work queue elements in a receive work queue at the destination endnode (pg 7, last ¶ continued on pg 8);
- storing state information in an end-to-end context at the source endnode and the destination endnode to ensure the reception and sequencing of message data sent from the source endnode to the destination endnode (Section 2.6 beginning on pg 9); and
- reliably sending data including performing a unicast of message mdata though the send work queue and the end-to-end context at the source endnode to the receive work queue and end-to-end context portion at the destination endnodes (Section 2.6 beginning on pg 9).

However, RFC 793 fails to disclose 1) reliable *multicast* to a group of destination endnodes wherein the reliable multicast comprise a series of replicated unicasts to each endnode and 2) generating a completion event to the source process in response to an indication that a *predetermined percentage of destination endnodes in the multicast group have reliably received a selected amount of message data multicast* from the source endnode.

With regard to point 1, reliable multicasting to a group of recipients through a series of replicated packets was well known in the art at the time of invention, as evidenced by Mockapetris. In a related art Mockapetris teaches a reliable multicasting method where the sender sends a separate (replicated) message to each destination endnode and receives an acknowledgement of receipt from each endnode separately (pg 153 Simulation algorithms, 1st ¶). Mockapetris further discloses that the sender maintains the multicast group of destination endnodes as a list (pg 153 Simulation algorithms, 1st ¶). It would have been obvious to one of ordinary skill in the art at the time of invention to add the multicast functionality disclosed by Mockapetris to the one-to-one transmission system disclosed by RFC 793 in order to provide a straightforward method for reliable one-to-many transmission (pg 153 Simulation algorithms, 1st ¶).

In the combined Mockapetris and RFC 793 system, all one-to-one transmission capabilities defined in the RFC 793 are expanded to a one-to-many (multicast) transmission capability since the one-to-many transmission is simply a series of replicated one-to-one transmissions to each multicast member endnode, as disclosed by Mockapetris (Cited above).

With regard to point 2, while RFC 793 discloses generating a completion event (TCP-to-user signals) for each endnode that acknowledges a received frame (pg 41, 5th ¶) neither RFC

793 nor Mockapetris specifically recited generating a completion event to the source process in response to an indication that a *predetermined percentage of destination endnodes in the multicast group have reliably received a selected amount of message data multicast* from the source endnode. Nonetheless it was widely known in the art at the time of the invention to notify application source processes that make multicast requests the status of such requests, as evidenced by Block. In a similar multicasting system, Block disclosed multicasting messages to a group of nodes (Col 9, lines 28-34). Like RFC 793 and Mockapetris, Block relies on acknowledgments from each node in the multicast group to determine which nodes have reliably received each multicast message (Col 15, lines 51-57 or Col 16, lines 8-11). In Block's system the source process that submits the multicast request can receive a completion event (notification) when a predetermined percentage of nodes (i.e. all nodes in the multicast group) have reliably received the multicast message (see inter alia Col 13, lines 40-43 and Col 16, lines 8-11, 18-19). This notification scheme allows the source process that submits a multicast request to be notified of the multicast request status. Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to incorporate the completion event notification scheme, as disclosed by Block, within the combined RFC 793 and Mockapetris system, so that application processes which use the combined RFC 793 and Mockapetris multicast system are apprised of the status of their multicast requests.

4. Regarding claims 3 and 26, RFC 793 discloses the source endnode including a network interface controller which packetizes the message data into frames (Section 2.2 beginning on pg 7).

5. Regarding claims 4 and 27, RFC 793 discloses the system wherein the destination endnodes each include a network interface controller which acknowledges receipt of frames sent from the source endnode (pg 6 Reliability section).

6. Regarding claims 5 and 28, RFC 793 discloses the system wherein the network interface controller and the end-toend context portion in the destination endnode ensures strong ordering of received frames sent from the source endnode, such that the frames are received in a same defined order as transmitted from the source endnode (pg 6 Reliability section).

7. Regarding claims 6 and 29, RFC 793 discloses the system wherein the source endnode retransmits frames that are not successively acknowledged in the reliable multicast service (pg 6 Reliability section).

8. Regarding claims 7 and 30, Mockapetris discloses the system wherein the network interface controller in the source endnode includes hardware which replicates frames to be provided in the series of unicasts (inherent, pg 153 Simulation algorithms, 1st ¶).

9. Regarding claims 8 and 31, Mockapetris discloses the system wherein the source endnode includes software verbs which perform the series of unicasts as a series of individual sequenced message send operations (pg 153 Simulation algorithms, 1st ¶).

10. Regarding claims 9 and 32, Mockapetris discloses the system wherein changes in composition of the endnodes participating in the multicast group are communicated to all endnodes participating in the multicast group (inherent, each sender maintains a list of multicast members and each member can be a sender; pg 153 Simulation algorithms, 1st ¶).

11. Regarding claims 10 and 33, Mockapetris discloses the system wherein the source endnode and each destination endnode maintains a list of destination addresses for all other endnodes participating in the multicast group (pg 153 Simulation algorithms, 1st ¶).

12. Regarding claims 11-12 and 34-35, RFC 793 discloses generating cumulative and per frame acknowledgments (pg 20, last ¶ continued on to pg 21).

13. Regarding claims 13-16 and 36-39, RFC 793 discloses gathering and counting acknowledgements from endnodes using a completion processing unit containing a completion queue (retransmission queue) (pg 21, first sentence below Functional Specification heading; pg 22 top ¶). RFC 793 further discloses informing the source process (through TCP-to-user signals) of an operation status of frames (pg 41, 5th ¶).

14. Regarding claims 17-18 and 40-41, RFC 793 discloses generating a completion event (TCP-to-user signals) for each endnode that acknowledges a received multicast frame (pg 41, 5th ¶). However, both Mockapetris and RFC 793 fail to teach generating a completion event when a certain percentage of endnodes have received the multicast frame. The Examiner takes Office Notice that it was well known in the computer networking art at the time of invention to generate events to a source process based on the percentage of completion (0% to 100%) for a given task. It would have been obvious to one of ordinary skill in the art at the time of invention to generate a completion event when a certain percentage of endnodes received the multicast frame in order to alert the host process of the multicast completion status.

15. Regarding claims 22 and 45, RFC 793 discloses the system wherein the completion processing unit includes a timing window, wherein expiring of the timing window without

necessary conditions for a completion event for a corresponding multicast frame occurring indicates that any missing acknowledgments art to be tracked and resolved (pg 10, 1st ¶).

16. Claims 19-21 and 42-44 are rejected under 35 U.S.C. 103(a) as being unpatentable over Request for Comment 793 (Transmission Control Protocol, hereinafter RFC 793) and Mockapetris (Analysis of Reliable Multicast Algorithms for Local Networks, Paul Mockapetris) and Block et al. (U.S. Patent Number 6,192,417; hereinafter Block), as applied to the claims above, and further in view of Aldrich (USNET post, John M. Aldrich Oct 16 1997).

17. Regarding claims 19 and 42, as discussed above RFC 793 discloses tracking ACKs from each endnode however, RFC 793 fails to teach using a bit-mask array for such tracking. Nevertheless, the use of bit-mask arrays to track events was well known in the art at the time of invention, as evidenced by Aldrich. In a related art, Aldrich discloses using a bit-mask array as a set of flags, which can be set and unset using bitwise operators (pg 2, top ¶). It would have been obvious to one of ordinary skill in the art at the time of invention to use a bit-mask array to track acknowledgements from endnodes in order to minimize memory consumption by only using a single bit to track each acknowledgement.

18. Regarding claims 20-21 and 43-44, RFC 793 discloses generating a completion event (TCP-to-user signals) for each endnode that acknowledges a received multicast frame (pg 41, 5th ¶). Further as discussed with respect to claims 2 and 25, Block disclosed generating a completion event when a certain percentage of endnodes have received the multicast frame (see inter alia Col 13, lines 40-43 and Col 16, lines 8-11, 18-19).

19. Claims 23-24 and 46-47 are rejected under 35 U.S.C. 103(a) as being unpatentable over Request for Comment 793 (Transmission Control Protocol, hereinafter RFC 793) and Mockapetris (Analysis of Reliable Multicast Algorithms for Local Networks, Paul Mockapetris) and Block et al. (U.S. Patent Number 6,192,417; hereinafter Block), as applied to the claims above, and further in view of Request for Comment 2236 (Internet Group Management Protocol, Version 2; hereinafter RFC 2236).

20. Regarding claims 23-24 and 46-47, while Mockapetris discusses maintaining a list of multicast members (pg 153 Simulation algorithms, 1st ¶) Mockapetris is silent as to how a multicast member list is updated. In a related art, the Internet Group Management Protocol Version 2 provides a protocol for maintaining multicast group membership (RFC 2236 pg 1, Abstract 2nd ¶) through the use of join (RFC 2236 pg 6, last line) and leave events (RFC 2236 pg 7, 1st bullet). It would have been obvious to one of ordinary skill in the art at the time of invention to incorporate the join and leave events taught by RFC 2236 within the combined RFC 793 and Mockapetris system in order to allow changes in group membership to be quickly reported to the multicast sender (RFC 2236, Abstract 1st ¶).

Response to Arguments

21. Applicant's arguments are noted however they are moot in view of the new grounds of rejection set forth.

Conclusion

22. The prior art made of record, in PTO-892 form, and not relied upon is considered pertinent to applicant's disclosure.

23. This office action is made **NON-FINAL.**

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Sean Reilly whose telephone number is 571-272-4228. The examiner can normally be reached on M-F 8-5.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Glen Burgess can be reached on 571-272-3949. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

11/18/2005

KRISNA LIM
PRIMARY EXAMINER